WE CLAIM:

Sub B2

~~1. A computer system providing type support for multiple type definitions, comprising:~~

~~an interface repository including:~~

~~a repository naming context; and,~~

~~a prefix naming context subordinate to the repository naming context, the prefix naming context serving as a root naming context for at least one interface definition language declaration.~~

~~2. The system of claim 1 wherein the prefix naming context further includes:~~

~~at least one naming context defined by an interface definition object and subordinate to the prefix naming context.~~

~~3. The computer system of claim 2 wherein at least one interface definition object has a fully scoped object name including a prefix name of the prefix naming context to which the interface definition object is subordinated.~~

~~4. The computer system of claim 1 wherein the prefix naming context is immediately subordinate to the repository naming context.~~

~~5. The computer system of claim 1 wherein the prefix naming context further includes:~~

at least one leaf node defined by an interface definition object.

6. The computer system of claim 1, wherein the prefix naming context is defined by a prefix object.

7. The computer system of claim 1, further comprising:

an interface repository loader that accepts as input parameters a specified interface definition language file containing at least one interface definition language declaration, and a specified prefix name, and installs the at least one interface definition language declaration in a prefix naming context having the prefix naming context in the interface repository.

8. The computer system of claim 7, wherein the interface repository loader creates a data file identified as related to the specified interface definition language file, and containing an identification of the specified prefix naming context.

9. The computer system of claim 7, wherein the interface repository loader creates the specified prefix naming context in the interface repository if the specified prefix naming context does not exist therein.

10. The computer system of claim 1, further comprising:

a memory device that stores the interface repository; and

a processing unit that executes operations of the interface repository loader.

11. The computer system of claim 10, wherein the processing unit further executes the interface repository loader to create a data file identified as related to the specified interface definition language file, and containing an identification of the specified prefix naming context.

12. A computer system providing type support for multiple type definitions, comprising:

at least one client object having a stub routine including a fully scoped name identifying a type providing an operation to the client object; and,

at least one server object having a skeleton routine including a fully scoped name identifying a type for the server object.

13. The computer system of claim 12, further comprising:

an interface definition language compiler that generates the stub routine in a client object.

14. The computer system of claim 12, further comprising:

an interface definition language compiler that generates a skeleton routine in a server object.

15. The computer system of claim 12, further comprising:

a memory device that stores the at least one client object, and the at least one server object; and,

~~a processing unit that executes a server object in response to an invocation of the server object by a client object.~~

~~16. The computer system of claim 15, wherein the processing unit executes an interface definition language compiler to generate the stub routine and the skeleton routine.~~

Sub B2

~~17. A method of providing type support for multiple type definitions, comprising the steps of:~~

~~defining in an interface repository a prefix naming context; and~~

~~storing the prefix naming context subordinate to a repository naming context in the interface repository, the prefix naming context forming an interface definition language root context for interface definition objects subordinate to the prefix naming context.~~

~~18. The method of claim 17, wherein each prefix naming context is stored immediately subordinate to the repository naming context.~~

~~19. The method of claim 17 further comprising the steps of:~~

~~specifying an interface definition language file containing at least one interface definition language declaration;~~

~~specifying a prefix naming context; and~~

~~storing each interface definition language declaration in the specified interface definition language file into the specified prefix naming context.~~

20. The method of claim 19, wherein the step of storing each interface definition language declaration further comprises the steps of:

creating an interface definition object for the interface definition language declaration;

storing the interface definition object in the specified prefix naming context; and

providing the interface definition object with a fully scoped object name including a prefix name from the prefix naming context in which the interface definition object is stored.

21. The method of claim 19, further comprising the step of:

creating a data file identified as related to the specified interface definition language file, and containing an identification of the specified prefix naming context.

22. A method of providing type support for multiple type definitions, comprising the step of:

providing an interface repository including:

a repository naming context; and

a prefix naming context subordinate to the repository naming context, the prefix naming context serving as a root naming context for at least one interface definition language declaration.

~~23. A method of providing type support for multiple type definitions, comprising the steps of~~

~~providing at least one client object having a stub routine including a fully scoped name for an object type providing an operation to the client object; and~~

~~providing at least one server object having a skeleton routine including a fully scoped name identifying an object type for the server object.~~

Sub B2

~~24. The method of claim 21, further comprising the step of:~~

~~providing a memory device that stores the at least one client object, and the at least one server object; and~~

~~providing a processing unit that executes a server object in response to an invocation of the server object by a client object.~~

add A2

add B2